macromedia® white paper

Macromedia MX: Strategies and Architectures for eLearning Content

by Tanya Heins and Frances Himes

September 2002

Macromedia, Inc.
600 Townsend Street
San Francisco, CA 94103
415-252-2000

Contents

This paper is an overview of how to best leverage Macromedia MX products to deliver Object-based eLearning content. The authors posit the advantages inherent in utilizing a dynamic delivery architecture that enables organizations to repurpose content at a highly granular level. A component-based architecture is delineated along with a definition for developing learning components that enable "templatizing" of instructional strategies. The final section of the paper is devoted to profiling a learning object demo application offered as a model for developers to consider when building their own eLearning application.

Macromedia focuses on enabling designers and developers to build effective communities of practice that support the implementation of innovative architectures, designs, and applications for eLearning. To this end, the company has produced three white papers aimed at various levels in an organization to help clarify issues central to eLearning. The first white paper describes the emergence of enterprise learning and Learning Objects and is intended for high-level decision makers within an organization.

This paper, the second in a series, describes the relationship between development strategies for leveraging Learning Object-based content and the related Macromedia technologies. It is intended to assist the eLearning manager who is developing a technology architecture or application for delivering Learning Object-based content. The third paper addresses the "how to" of implementation. It is addressed to the development team involved in building eLearning solutions.

Accompanying these three papers is a Learning Object demo application that provides an example of the potential implementation of Macromedia MX tools within Learning Object architectures. The Macromedia MX eLearning architecture is built on the concept of dynamic Learning Objects that allow for a separation of the data, logic, and presentation layers. This simplifies the creation, use, and reuse of Learning Objects that can be leveraged for use in a variety of eLearning contexts and for delivery on multiple devices.

Strategies for eLearning Development

Dynamic web technologies - that can leverage multiple systems and databases of information - are gaining wider adoption in campus and corporate systems today. Unlike the static content, which uses the server as a storage device for an entire file and must be edited individually in the appropriate software application, Dynamic pages separate out the logic, data, and presentation layers, which allows presentation and learning logic to be authored once on the server and used in many instances for delivery to learners. Server applications, such as Macromedia's ColdFusion MX and Macromedia's Flash Communication Server support delivery of dynamic content. Development tools such as Macromedia Flash MX and Dreamweaver MX support the dynamic content approach through a component-based development environment that allows server or interface programmers to share and reuse programming logic.

Separating out the data makes it is possible to build browser-based editing interfaces that can be used even by persons without technical programming knowledge. Dynamic learning content delivery also allows an organization to build a single consistent interface for the presentation of Learning Objects, from which greater efficiencies of scale in production and delivery result.

This white paper demonstrates the value of a component-based development approach, explains how Macromedia's MX tool set can be used to build, manage and store eLearning content, and examines how to use MX Studio, ColdFusion MX, and Flash Communication Server for eLearning content delivery.

eLearning managers will learn how Macromedia MX tools can be used for the assembly and delivery of Learning Objects through the use of templates that can be adapted to an organizations instructional model. To aid this discussion, a sample application of MX tools is introduced in an eLearning Demo Application that focuses on the development of learning objects. (http://www.macromedia.com/resources/elearning/objects) The Macromedia MX tools allow us to:

- Improve the technical development team workflow through the use of components.
- Leverage instructional strategies for developing Learning Objects.
- Allow non-technical Learning Object (LO) developers to easily assemble and edit LOs through the use of a browser.

In the Learning Object Demo example, discussed in more detail towards the end of this paper, Flash MX is used as the Rich Client[i] User Interface for delivery of Learning Objects.

The Learning Object

The eLearning community's approach to Learning Objects is actually based upon an earlier model of Object-oriented programming that has been around since the 1960s[ii]. For the purposes of our discussion here, a Learning Object (LO) will be defined as a unit of instructionally sound content centered on a learning objective or outcome intended to teach a focused concept. A LO may contain opportunities for practice, simulation, collaborative interaction, assessment, and educational resources.

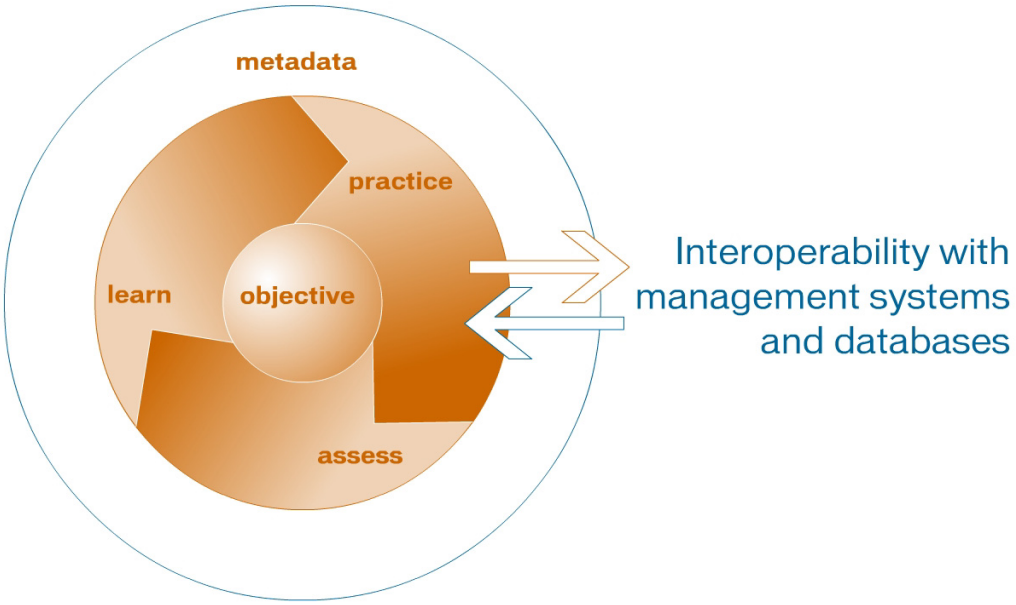

Figure 1: *Structure of a Learning Object*

A LO is constructed from Media Assets, such as paragraphs of text or html, screen titles, captions, video, animation, diagrams, and sound narration. Media Assets can be produced and managed with tools available in the MX Studio.

Static and Dynamic Learning Object Design

As previously mentioned, Static content uses the server as a storage device for an entire file. For any changes to be made to static content, someone proficient in the appropriate software application must edit it individually.

Dynamic pages separate out the logic, data, and presentation method, which are then assembled as required by server applications. ColdFusion MX and Flash Communication Server are examples of server applications. With the data separated out from the logic and presentation, it is possible to build an editing interface that allows users without technical programming knowledge to contribute to the development of rich and engaging LOs.

To illustrate this concept, consider the following: Common HTML webpages are made up of static content. Pages are created and stored as separate, stand-alone files. Editing these HTML pages is accomplished by opening the file up in an editing program, making the appropriate changes, and then saving the file back to a web server. All the presentation, content, and logic that determine how the page works are contained within this one stand-alone file (as shown in Figure 2, which illustrates a simple HTML webpage).

```
<html>
<head>
<title>Static Web Page Title</title>
</head>
<body>


<img src="images/logo.gif" width="158" height="78" alt="Company Logo">


<h1>This is the static heading</h1>


<p>This is the static objective. Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat.</p>


<p>This is the static content. Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud
exerci tation ullamcorper suscipit lobortis.</p>


<p>This is the static assessment. Lorem ipsum dolor sit amet, consectetuer
adipiscing elit.</p>


<p>Contact Information: Instructor's Name, email@instructor.com,
222.444.7878, 8am - 10am EDT.</p>


</body>
</html>
```

Figure 2: *Static HTML webpage*

Static web content design and delivery can be easy to build and maintain on a small scale, but the task becomes more and more difficult as the scale increases. Imagine for a moment what might be required to edit a small section of content that appears in three different files. Then consider what might be required to edit that same section of content in three hundred files. Each and every edit within static content architecture requires individual attention. This proves to be an issue where scalable course content is required.

In contrast, Dynamic web content offers a more flexible architecture wherein content, logic, and presentation can be designed as separate components. This process allows developers to mix and match depending upon the specific context. Edits in a single content object result in corresponding changes wherever that object is used. Dynamic learning content delivery, however, does— at least initially—require specific skills and resources to implement.

The following code (in Figure 3) takes the previous HTML webpage (from Figure 2) and transforms it into a dynamic webpage. In a sense, this page now becomes a "template" into which different content may be placed from an external database source.

```
<cfinvoke
component="components.LearningObject"
method="LOstructure"
returnvariable="LOstructureRet">
<cfinvokeargument name="LO_ID" value="#URL.LO_ID#"/>
</cfinvoke>


<html>
<head>
<title><cfoutput>#LOstructureRet.LO_Title#</cfoutput></title>
</head>
<body>


<img src="images/<cfoutput>#LOstructureRet.LO_Logo#</cfoutput>"
alt="<cfoutput>#LOstructureRet.LO_LogoAltTag#</cfoutput>">


<h1><cfoutput>#LOstructureRet.LO_Heading#</cfoutput></h1>


<p><cfoutput>#LOstructureRet.LO_Objective#</cfoutput><p>


<p><cfoutput>#LOstructureRet.LO_Content#</cfoutput><p>


<p><cfoutput>#LOstructureRet.LO_Assessment#</cfoutput><p>


<p><cfoutput>#LOstructureRet.LO_contactinfo#</cfoutput><p>


</body>
</html>
```

Figure 3: *Dynamic ColdFusion webpage*

This single web page template may now be used any number of times to display a variety of content, by simply passing a unique identifying variable to the page. For example, a course web page is passed the variable or ID for Learning Object (LO) 34 - as it loads into the learners' web browser. The dynamic web page can request all of the relevant information in the database table for record number 34. By passing a different variable, 27, for example, the learner would then see the content for that new LO.

Figure 4: below further illustrates the concept of separating out the data, logic and presentation layers into specific component-based technologies. By grouping elements from each layer we can re-assemble Learning Objects without duplicating media asset development or programming work. In this diagram metadata and media assets, a CFC web service that packages content based on the SCORM model, along with UI device display parameters and UI components are packaged together and presented as a Learning Object.

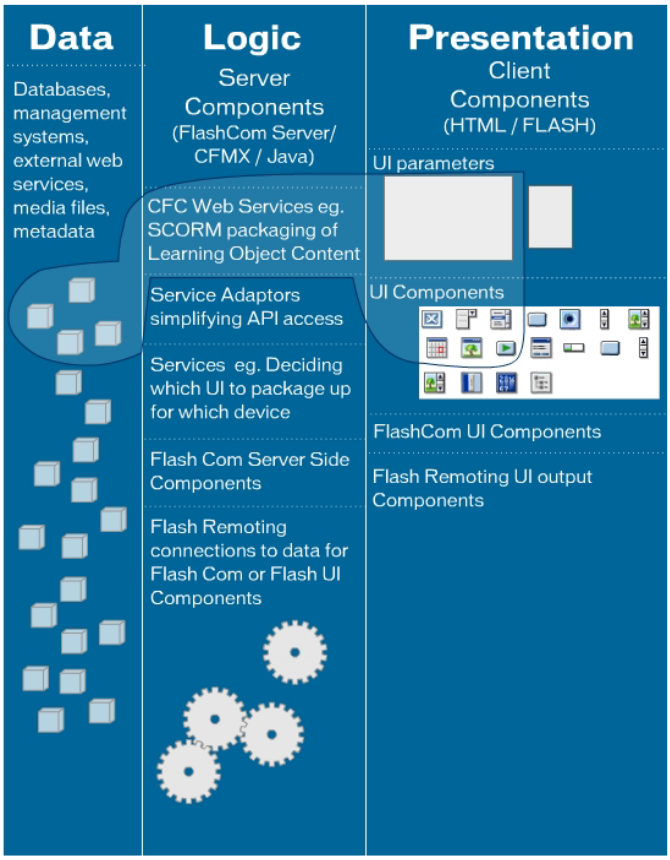

Figure 4: *Aggregation of data, logic and presentation to dynamically deliver content*

Dynamic learning content delivery also allows an organization to build a single consistent interface for the presentation of LOs. Granted, one single presentation format will not fit all situations. However, it will significantly cut down on the amount of development work needed to produce LOs. By "templatizing" instructional strategies wherever possible, greater efficiencies of scale in production and delivery will result.

The HTML examples (Figure 2 and 3) above use a simplified web page to illustrate the difference between static and dynamic content. More sophisticated methods of delivering learning content and LOs can involve the use of a Macromedia Rich Client Flash MX interface. For example, presentation layers that accommodate the screen scale and appropriate changes in navigation logic for delivery to multiple devices, can allow organizations to deliver content on the fly without individual authoring to adjust each learning object for delivery on multiple devices. This concept of can be extended to the delivery of content in multiple contexts [Figure 5].

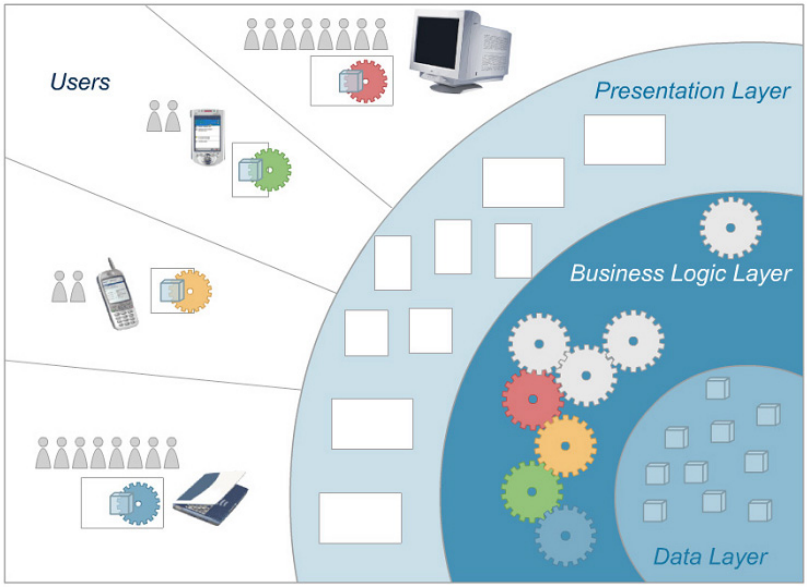

Figure 5: *Aggregation of data, learning logic and presentation to multiple devices*

Dynamic Flash interfaces can also make use of additional technologies, like Flash Remoting and XML, to handle interoperability with databases and other systems.

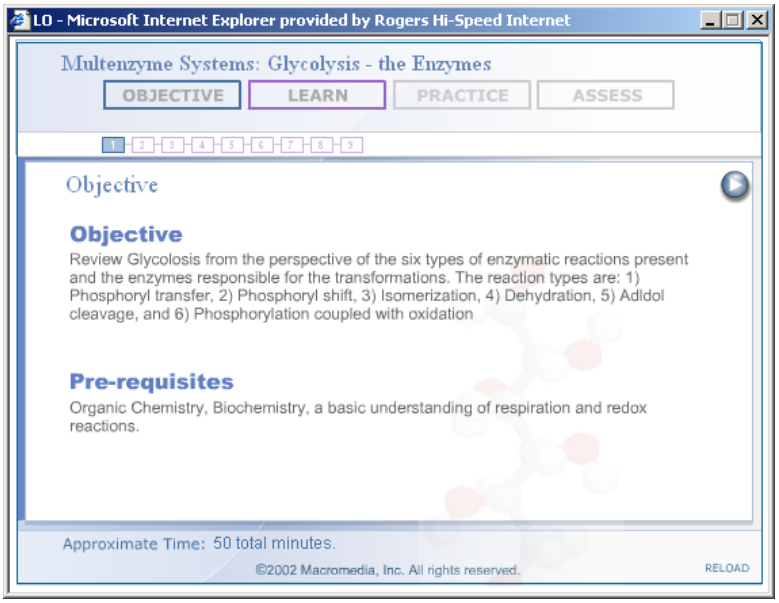

Figure 6: *Flash MX used as a rich client interface for Learning Object delivery*

Dynamic web technologies are gaining wider adoption in the campus/corporate systems. The use of databases and open integration into other systems, such as those used by academic registrars or accountants, demonstrates how eLearning has shifting its static content architecture to a more scalable dynamic model. East Carolina University for example, uses ColdFusion to create an innovative "one-stop" university portal, which brings just-in-time information and services to thousands of students, faculty, and staff[iii].

Advantages and Disadvantages

For the design and development of eLearning content, let us consider this summary of advantages and disadvantages of the static and dynamic approach:

	Static	**Dynamic**
Creating single Learning Object	Create an LO through custom assembly of Media Assets.	Requires creation of general template first, then content for the LO.
Scalability of project/application	Create each LO separately.	With the template built, only the actual content for each LO needs to be created, provided all LOs use the same template or instructional strategy.
Real-time synchronization with database	Not applicable.	Dynamic LO development is an obvious solution.
Maintenance	High Maintenance. Adding new LOs means creating	Low Maintenance. New content is inserted into an already built

	separate LO files. Any change in the common structure (e.g., when updating a menu item that appears on all LOs) has to be propagated to every static file, requiring manual updates to each.	learning component template making any editing and additions a snap. Any changes to the delivery structure mean updating only the template.
Learning curve	Low. Requires basic HTML or Flash animation knowledge.	Requires more programming knowledge, database development experience, but less overall administration once the dynamic delivery model is in place.
Offline Delivery	Easily packaged for CD-ROM delivery or cached in the browser for later review.	Dynamic data requires live internet connection to access latest version of information stored in database. A blended approach to generating static files is an alternative if partial offline delivery is a consideration.

Source: adapted from fumot.com's Building Dynamic Websites[iv]

Learning Components and the Development Process

What happens if we extend the concept of modular, reusable LOs to the development process itself? The result is an object-oriented approach featuring reusable packets of code, called components. These become the building blocks for the development process.

Components

Components refer to any piece of code intended for reuse by a developer within a given application or across multiple application environments. A component may range in size from a small, simple client component (e.g., the user interface button control) to more complex server components, like those that handle logic for variables such as user roles. A component, in turn, may be made up of other smaller components.

The integrated component-based system within the Macromedia MX suite of tools helps a developer create and re-use existing parcels of code. One particular benefit in tight product integration is that each MX tool can connect to components in other MX tools. This awareness helps streamline the workflow between application developers, instructional designers, subject matter experts, and interface developers—each of whom is likely working in a separate, but connected, integrated development environment (IDE)[v].

The component workflow method has an added advantage in that it allows an organization to effectively leverage the expertise of highly skilled developers. Components developed by experienced developers can easily be implemented in production by lesser skilled developers. By establishing libraries of reusable components, an organization can more efficiently manage, consolidate, and track the development processes and code assets.

Another advantage is that it allows one to isolate expertise. Developers can program flexible, reusable components. Content experts or instructional designers can provide the instructional design and content as media assets through a web interface that uses components and allows them to create LOs without technical knowledge or programming skills. The ability to use expertise when appropriate without requiring that faculty subject matter experts become programmers is paramount when considering the accessibility of creating LOs.

Learning Components

Components, when grouped and organized into an instructionally sound structure, are called Learning Components. These Learning Components might include simple teaching facts, concepts, or principles or more complex strategies, such as case studies based on live data, expert-led workshops, mentored exercises, and role-based simulations. Learning Components can also be developed to adapt to individual learning needs.

Learning Components can be used in either a static or dynamic manner. A collection of client and server Components bound together as a Learning Component can provide a complete, deployable framework that allows for the reuse of eLearning delivery methodologies. Organizations can leverage development efforts by using a single Learning Component per instructional strategy. This Learning Component could be constructed to *dynamically* generate many LOs, each with different content during run-time, or simply used in the development process to produce a *static* LO. The eLearning Demo Application discussed on page 16 uses *dynamic* Learning Components.

An Architecture to Support Learning Components

> *"Most web applications today are developed without significant structure. Often they consist of server-side dynamic pages with embedded script, presentation logic and data access logic. As a result, most of the logic components and data associated with these applications are locked up—unable to be shared inside and outside of corporations. In essence, web applications and data are kept in silos, limiting the ability of companies to share their information and business logic." Jeremy Allaire*[vi]

In order to implement scalable Learning Components, a specific technical architecture is necessary. This tiered architecture separates data, logic, and presentation, and allows for adaptive and dynamic assembly of components from each of these layers to present eLearning content.

Greater efficiencies of production and distribution result from this type of tiered architecture. Separating data, logic, and presentation makes it possible to eliminate redundancies in the system. With careful planning and co-ordination, a component-based development approach for eLearning applications and LOs will provide an efficient, scalable solution that will lend itself to sharing eLearning content, once technologies like web services become prevalent.

The illustration below shows how components can be tiered into the data, logic, and presentation layers that are assembled as necessary for delivery to the browser. The MX studio provides the developer with tools capable of building this type of architecture.

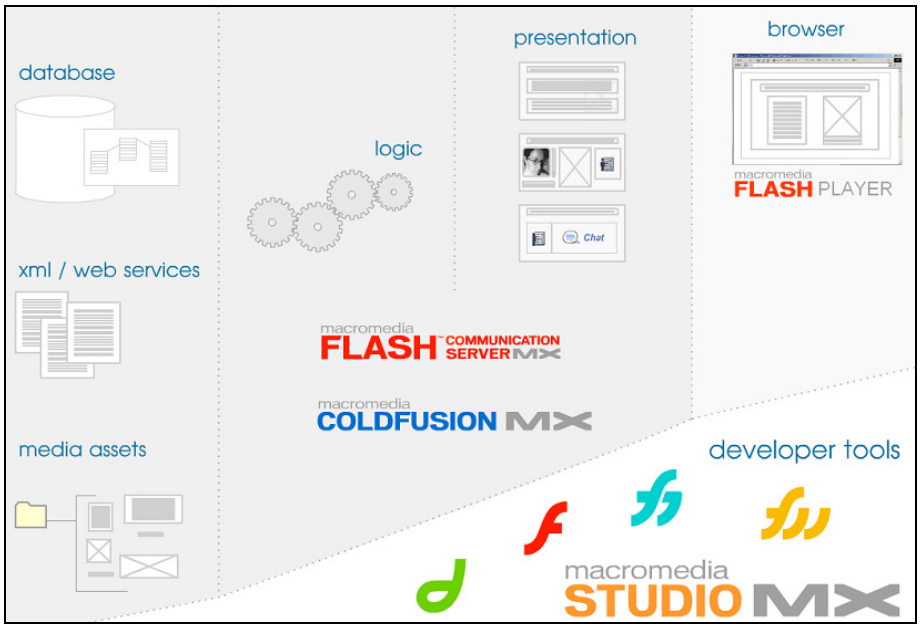

Figure 7: *Use of MX products in tiered architecture*

Learning Object Demo Example[vii]

The Learning Object Demo application uses Macromedia MX tools to: improve a technical development team's workflow through the use of components, to leverage instructional strategies for developing LOs, and to allow non-technical LO developers to easily assemble and edit LOs through the use of a browser.

Macromedia Flash MX is the Rich Client User Interface for LOs, Flash Communication Server manages the collaborative communication present in the LOs, and Flash Player allows the student-user to experience the LOs through the browser.

Instructional Designers can easily assemble the LOs through a web application built in ColdFusion MX. Media Assets uploaded and managed by the CFMX application are stored in a database or on the server. CFMX publishes a XML document for the LOs created. This XML document is then used by Flash MX to dynamically structure and assemble an LO from Media Assets.

Separation of the interface, logic, and data allows for maximum reuse of the component parts. The "glue" used to bind communication between the interfaces, logic, and data is SQL, XML, and FlashRemoting.

Creating a Learning Object Using the Demo Application

An Instructional Designer or Content Expert can use his/her browser to design and create an LO. He/She would start by filling out a form to create an Objective upon which the LO content would be based. In this application, there is also an opportunity to provide information about pre-requisite skills necessary to complete the LO.

This interface is a ColdFusion MX application that is uses Components to send information to the database or server.

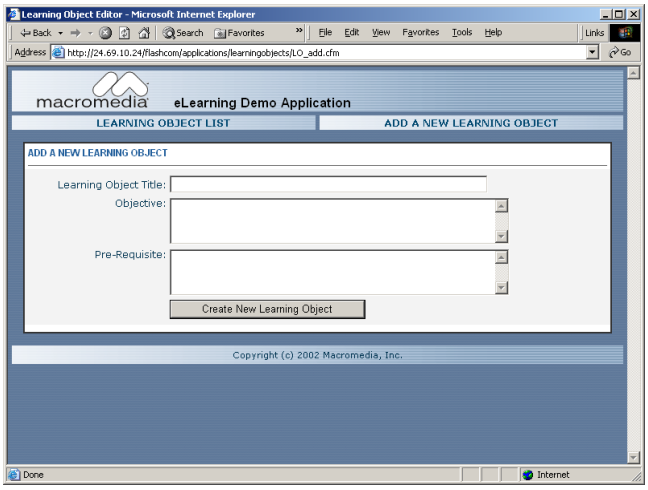

Figure 8: *Creating a Learning Object*

Choosing Learn, Practice, and Assess Screens

Once the Learning Objective form is submitted, the user can set about choosing appropriate screen layouts for the presentation of learning content. A Developer can customize these screen layouts or learning templates to suit different instructional strategies. As each screen is chosen, it is designated as either a "learn," "practice," or "assess" screen. Below is a table of some of the layouts included in this demo application:

	Placeholders for a title at the top, a section of text at the left and a graphical media asset to the right with a caption below.
	Placeholders for a title, video, and audio feed for an instructor at the left. This video can be recorded for asynchronous playback. In the center is an area for a graphical media asset. This could be a simulation, interactive diagram, or simply a graphic. To the right is a list of students who are currently online.
	This layout has a title and collaborative element to allow users to create an online slide show.
	The placeholder for the title appears at the top. Below, to the left is a chat application that can record the discussion. To the right is a shared whiteboard, to which users may contribute. Below the transparent background of the whiteboard is a placeholder for a graphical media asset. This provides an opportunity to present an element that might be the subject of a whiteboard discussion.

Figure 9: *Screen Layouts for creating a Learning Object*

Adding Media Assets to Screens

Once the appropriate screens have been determined, the Instructional Designer or Content Expert can upload the appropriate Media Assets from their local workstation to the database and server. The figure below shows the simple form that allows users to locate files locally and to upload them for use as Media Assets.

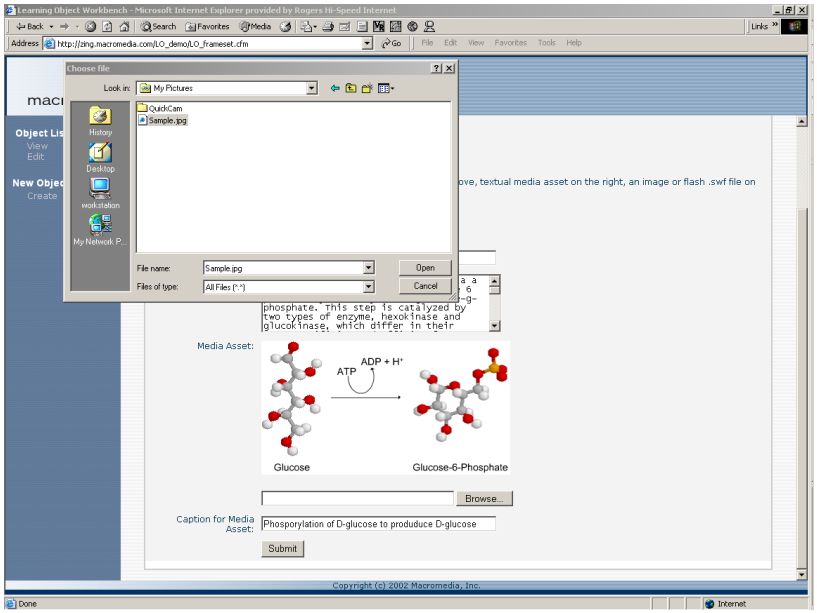

Figure 10: *Adding Media Assets to a Screen*

Reviewing the Learning Object

After the LO is assembled in the "workbench" area of the demo application, it is possible to click on a "preview" link to see what the LO would look like to a learner. The LO still needs to be contextualized within a delivery context. Learning and content management systems that allow open-source communication, through APIs or XML for example, could be integrated with the Demo application and use the assembled LOs to contextualize the assembled LOs in a course, for a performance support tool, or perhaps for a just-in-time search of an organization's repository of LOs.

Extending the Demo Application

Though not within the scope of this Demo application, external management systems and databases can be integrated through Macromedia tools. MX Studio applications can use data exchange methods such as SQL, XML, URL encoding, and Flash Remoting to communicate with other systems. For detailed information on how the Demo Application functions, refer to the third white paper: **Macromedia MX: Creating Learning Objects[viii].** The third white paper outlines the anatomy of a learning object, offers design and development strategies, and suggests technical and delivery best practices for building and deploying learning objects using the suite of Macromedia MX products.

Conclusion

The path to utilizing Object-based design as a development model to create interactive learning environments is possible with today's technologies. Challenges remain in the development of definitive standards and effective taxonomies and in content development and reuse. As eLearning becomes more essential to productivity in education, industry, and government, unified efforts will be needed to leverage existing and future technology investments to move the eLearning industry toward development and delivery models that demonstrate efficacy in learning, productivity, and performance.

To continually evolve and demonstrate the best practices in eLearning, Macromedia welcomes feedback from its customers.

Macromedia Learning Objects Development Center

Macromedia is committed to help businesses, educational institutions, and government agencies create personalized, relevant learning experiences. The Macromedia Learning Objects Development Center hosts a series of whitepapers, development assets, and implementation models that will help organizations leverage their technology investments to improve organizational and employee productivity and performance. Visit www.macromedia.com/go/objects to access the full set of resources available.

Tanya Heins *is an eLearning consultant with a background in teaching, fine arts, and applied research in eLearning. She holds an M.F.A from the University of Victoria in British Columbia, and has had more than eight years of professional interactive design and development experience, ranging from designing CD-ROMs and database-driven eLearning to custom development of award winning learner and content management systems.*

You can contact Tanya at tanya@numen.ca

Frances Himes *is the Associate Vice President of Higher Education at Macromedia, in which position she manages higher education business and contributes to Macromedia's eLearning product strategy. Over the course of her career her work has focused on learning object design and development, adaptive eLearning architecture and the development of contextual eLearning environments. She has also developed curriculum for teaching intercultural forms of learning, and teaching and learning with technology. She has served at the University of Arizona, as both the co-author and Project Director for the Virtual Adaptive Learning Architecture Project and as Director of Global Programs, at Western Michigan University, as Director of Research and Development for Information Technology, and at the University of Wisconsin-Eau Claire, as the Associate Director of International Education.*

You can contact Frances at fhimes@macromedia.com

[i] "Rich Clients and the Internet Experience" by Jeremy Allaire. Available at: http://www.macromedia.com/software/mx/info/rich_client.html

[ii] "The History of Simula" by Jan Rune Holmevik. Available at: http://java.sun.com/people/jag/SimulaHistory.html

[iii] Higher Education Showcase: East Carolina University. Available at: http://www.macromedia.com/resources/education/showcase/archive/ecu.html

[iv] Original table can be found at http://www.fumot.com/techtalk/Dynamic/introb.htm

[v] "The Design of Dreamweaver MX" by Sho Kuwamoto. Available at: http://www.macromedia.com/desdev/mx/dreamweaver/articles/dw_design.html

[vi] "Macromedia Flash MXA next-generation rich client" by Jeremy Allaire. Available at: http://www.macromedia.com/desdev/mx/flash/whitepapers/richclient.pdf

[vii] This demo can be viewed at http://www.macromedia.com/resources/elearning/objects and details on how to build such an application can be found in the third white paper: Macromedia MX: Creating Learning Objects

[viii] The collection of three related white papers can be found at: http://www.macromedia.com/resources/elearning/objects